

함수

정의

일련의 과정을 문 으로 구현하고 코드 블록으로 감싸서 하나의 실행단위로 정의

매개변수 : 함수 내부로 입력을 전달받는 변수

인수 : 입력

반환값 : 출력

장점

1. 유지보수의 편의성 상승
2. 코드의 신뢰성 상승
3. 코드의 가독성 상승

일반 객체와의 차이

1. 일반객체는 호출이 불가하지만 함수는 호출할수 있다
2. 일반 객체에는 없는 함수 객체만의 고유한 프로퍼티를 가진다

함수의 리터럴

구성요소	설명
함수 이름	<ul style="list-style-type: none"> - 함수 이름은 식별자다 - 함수이름은 함수 몸체 내에서만 참조할 수 있는 식별자다. - 함수 이름은 생략할 수 있다.(익명함수)
매개 변수 목록	<ul style="list-style-type: none"> - 0개 이상의 매개변수를 소괄호로 감싸고 쉼표로 구분 - 각 매개변수에는 함수를 호출할 때 지정한 인수가 순서대로 할당된다. - 매개변수는 함수 몸체 내에서 변수와 동일, 식별자 네이밍 규칙을 준수 해야한다.
함수 몸체	<ul style="list-style-type: none"> - 함수가 호출되었을 때 실행될 문들을 하나의 실행 단위로 정의한 코드 블록 - 함수 호출에 의해 실행된다.

함수의 정의

※함수를 정의하는 4가지 방법

함수 정의 방식	예시
함수 선언문	<pre>function add(x,y){ return x+y; }</pre>
함수 표현식	<pre>var add = function(x,y){ return x+y; };</pre>
Function 생성자 함수	<pre>var add = new Function('x','y','x+y');</pre>
화살표 함수(ES6)	<pre>var add = (x,y)=>x+y</pre>

함수의 선언

함수 선언문은 함수 이름을 생략할 수 없다. 자바스크립트 엔진은 함수 선언문을 해석해 함수 이름과 동일한 이름의 식별자를 암묵적으로 생성하고, 거기에 함수 객체를 할당한다. 반면 함수 표현식(함수 리터럴)은 “함수 이름은 함수 몸체 내에서만 참조할 수 있는 식별자”라는 특성이 있어, 별도로 변수에 할당받지 않은 이상 호출할 수 있는 식별자가 없다.

```
// 기명 함수 리터럴을 단독으로 사용하면 함수 선언문으로 해석된다.
// 함수 선언문에서는 함수 이름을 생략할 수 없다.
function foo() { console.log('foo'); }
foo(); // foo

// 함수 리터럴을 피연산자로 사용하면 함수 선언문이 아니라 함수 리터럴 표현식으로 해석된다.
// 함수 리터럴에서는 함수 이름을 생략할 수 있다.
(function bar() { console.log('bar'); });
bar(); // ReferenceError: bar is not defined

// 아래와 같이 작성해야 호출 가능하다.
bar = (function bar(){ console.log('bar'); });
bar(); // bar
```

암묵적 생성

함수 객체를 가리키는 식별자가 없으면 참조할 수 없으므로 생성된 함수를 호출하기 위해 함수 이름과 동일한 이름의 식별자를 암묵적으로 생성한다. 그 후 거기에 함수 객체를 할당한다.

함수는 함수 이름으로 호출하지 않고 식별자로 호출한다.

일급 객체

값의 성질을 갖는 객체를 일급 객체라 명한다.

함수는 일급객체로서 값처럼 자유롭게 사용가능하다.

호이스팅

함수 선언문으로 함수 선언시 함수 호이스팅

함수 표현식으로 함수 선언시 변수 호이스팅

매개변수

순서에 의미가 존재한다.

최대한 적게 만들어서 함수가 한가지 일만 하게하는것이 효율적이다.

부수효과

객체타입 인수는 함수 몸체에서 객체를 변경시 원본이 훼손된다.

이를 방지하기 위해서는

1. 옵저버패턴
2. 불변객체

등을 사용하여 부수효과를 방지해야한다.

함수의 종류

즉시 실행 함수

함수를 선언과 동시에 즉시 호출되는 함수이다.

단 한번만 호출되며 다시 호출할 수는 없다,

재귀함수

함수 내부에서 자기 자신을 호출하는 함수

재귀함수는 자신을 무한 재귀 호출하기 때문에 탈출조건을 반드시 생성해야한다.

콜백함수

함수의 매개변수를 통해 다른 함수 내부로 전달되는 함수

받는 함수는 고차 함수가 되어 콜백함수를 자신의 일부분으로 합성한다.

순수,비순수 함수

부수효과가 없는 함수를 순수함수

부수효과가 있는 함수를 비순수함수라 칭한다

당연히 부수효과가 적은 순수함수가 더 좋다.

참고

- <https://hong-p.github.io/javascript/javascript-deepdive-ch12/>